THE JOURNAL OF NAVIGATION (2010), **63**, 323–341.
doi:10.1017/S0373463309990385

Integration of a Geographic Information System and Evolutionary Computation for Automatic Routing in Coastal Navigation

Ming-Cheng Tsou

(*National Kaohsiung Marine University*, *Taiwan*)
(Email: d86228006@yahoo.com.tw)

Suitable route planning is related to the safety and economy of navigation. However, route planning has become increasingly complex over the years and the planning process requires a large amount of oceanic environmental information. In order to use the oceanic environmental information effectively and improve the efficiency of route planning, this research employed a Geographic Information System (GIS) as the platform for enabling two-phase automatic route generation design. Firstly, through GIS's spatial data management, spatial analysis and geometric computation capability, the presence of the obstacle is detected and candidate routes are automatically generated. These are provided to the evolutionary algorithm as the basis for preliminary population calculation. Then, a specially designed evolutionary algorithm is used for route elimination to obtain the optimal route, resulting in the most-recommended routes that encompass safety and economy. This technique is more efficient than evolutionary computation techniques that use traditional random searches. At the same time, this targets safety and economy, providing a reference for developing a route planning strategy.

KEY WORDS

1. Genetic algorithm. 2. Routing. 3. Obstacle avoidance. 4. GIS.

1. INTRODUCTION.

Marine transportation forms the main pillar of international trade. However, while automated ships are common these days, problems such as marine collisions and groundings occur often. These problems not only result in loss of life and property loss, but they also create serious disasters for the marine ecosystem and the environment. It has been suggested that 85 percent of all marine collisions and groundings are due to human error (Dove, *et al*., 1986). In marine grounding cases, 22 percent of the cases are due to inadequate route planning (Hayashi & Kuwajima, 1991). In particular, an optimal route plan has a huge impact on ship navigation; if a route that avoids danger zones can be thoroughly planned prior to sailing, the number of navigational accidents can be lowered and navigational safety thus improved. Apart from the direct benefit to ship navigational safety, a good route plan can also save fuel and the additional costs arising from accidents. Furthermore, as the global warming problem becomes more serious, reducing fuel consumption is a topic that concerns the whole world and is an active globally targeted goal. Apart from pollution of the oceans, air pollution caused by the use of heavy fuel oil in marine combustion engines is a growing problem. Therefore, an additional option to reduce air pollution is to decrease fuel consumption during the transit (Bijlsma, 2008). In order to realize more economic, safer ship navigation and a cleaner ocean, researchers have conducted a wide range of works in automated ship collision avoidance through the use of qualitative analysis techniques, quantitative analysis techniques, a combination of the two types of techniques, and engaging in information technology applications such as expert systems (Coenen, 1980), neural networks (Zhu, *et al*., 2001), artificial intelligence (Lee, *et al*. 2004, Kao, *et al*., 2007, Park, *et al*., 2007) and decision support systems (Jones, 1978). Nevertheless, although research that addresses the problem of automatic generation of an obstacle avoidance strategy in wide waters has achieved considerable results, the same problem for confined waters has not been addressed widely. The main reason is that the problem of obtaining reliable hydrographic results and electronic chart information sources has not been addressed (Yang *et al*., 2007) and that there is still a great degree of reliance on paper charts for route planning.

In the past, when route planning was performed on paper charts, the navigator typically first plotted the route manually on the chart, manually decided if a route was applicable and adjusted the route accordingly. In this form of manual route planning, the actual chart information may be concealed due to obstacles that are overly small or a map that is damaged, resulting in decisional errors based on the given information. Consequently, this leads to adverse results in navigation. Besides this, manual route design is heavily dependent on the navigator's experience and operating skills, which often brings a sense of unease in respect of the safety of the navigation. Following the rapid development and application of Electronic Chart Display and Information System (ECDIS), information related to the oceanic environment can be integrated to a single platform, providing an essential tool for route design. Using ECDIS to design navigation routes, the system can automatically determine the presence of shallow water below the point of safe-depth, the presence of isolated hazardous objects, whether the ship has exceeded the safe-depth isobaths and whether it will pass through danger zones. This prevents user operational errors. However, at present, the route planning feature provided by ECDIS operates in the same way as paper charts, which require the navigator to input waypoints manually. Specifically, ECDIS provides only a platform for assisting in route design and error checking. This does not fully utilize the intelligence capacity of ECDIS and an optimum route plan still requires additional design work by an experienced navigator.

This research builds a model of operation and display that is based on the GIS platform, on top of the ECDIS, to enable a two-phase automated route generation design. First, route obstacles are determined through GIS's spatial data management, spatial analysis and geometric computational capability. Then, multiple candidate routes are automatically generated and these are used as preliminary input routes to the evolutionary algorithm. The decision on a safe obstacle avoidance route is a multi-criteria, non-linear planning problem which contains a large number of constraints, and a suitable balance needs to be achieved between the safety and economy of the route (Smierzchalski and Michalewicz, 2000). This is to ensure that the avoidance strategy undertaken not only maintains route safety by using risk assessment and avoidance measures, but also minimizes deviation from original routes when avoiding obstacles. In order to address these goals, we employ a popular form of artificial intelligence in the field of evolutionary algorithms, the genetic algorithm (GA), which models biological evolutions, to propose the fitness function, gene coding and genetic operation for suitable collision avoidance route generation. It is hoped that, through the design in this research, the navigator will simply need to input the latitude/longitude coordinates and related parameters of origin and destination to automate route generation within the system. Furthermore, this would use a better designed, more objective, faster and accurate model to automatically generate a route that is more reasonable, safer and more economical, so as to reduce the navigator's workload and provide a reference for route planning.

2. PREVIOUS WORK.

Previous related route planning literature can be divided into two main areas of study. The first is weather routing in trans-oceanic seafaring. This form of research covers a wider area, normally covering the Atlantic Ocean or Pacific Ocean and mainly considers the effects of short- to mid-term weather: in particular, the effect of wave direction and wave height on the speed of the ship across the ocean. Since the appearance of an innovative paper by Hanssen and James (1960), several methods have been proposed to solve the problem of minimizing sailing time. The main methods include the isochrone method, calculus of variation, optimal control theory and dynamic programming. As this form of research involves crossing the ocean, the works here do not consider obstacle avoidance in the route and the main focus is weather routing, pursuing a shorter sailing period or lower fuel consumption as the sole target. Nevertheless, there are many areas worth investigating in route planning methodology. For instance, in Bijlsma (2001, 2008), application of the calculus of variations was extended to the minimization of fuel consumption; in Bijlsma (2002) the relation between optimal control theory and dynamic programming was elucidated for the case of minimal fuel routing, and in Bijlsma (2004) this was done for a ship with limited manoeuvrability. Szlapczynska and Smierzchalski (2007) revised the isochrone method using computerized calculations to generate the preliminary candidate route population for evolutionary computation, as a means of improving the efficiency of evolutionary computation when randomly generating the preliminary population.

The second area of research concerns obstacle and collision avoidance route planning in short- to mid-distance coastal navigation: this form of research is derived from robot collision avoidance techniques and emphasizes avoidance path planning for dynamic and static objects within a short distance. Here, a lot of the research uses the geometric approach to resolve the collision avoidance problem, seeking the shortest obstacle avoidance route. As the sole target is to seek the shortest route, the route chosen may not be the best route and the generated route may not comply with navigational practice. Thus, they are not suitable for use in route planning that considers multiple goals. Hayashi and Kuwajima (1991) employ a fully automatic stranding avoidance system that uses radar matching to obtain positions with high precision, and to make judgments concerning the danger of stranding. Harris *et al.* (1999) used a neuro-fuzzy network multiple ahead predictor model on ship steering dynamics, with rudder deflection angle as system input and ship directional angle as system output using a waypoint guidance scheme based on line-of-sight. Yang (2007) and Zhang *et al*. (2007) use a special search method to discover dangerous static obstacles and, through geometric computation, automatically generate an obstacle avoidance route. Their technique is suitable for short distances and simple avoidance situations for obstacles. However, when applied to mid-distance route planning, and when the obstacle avoidance situations are more complex in certain navigation situations, their technique might not be suitable. Lee (1961) and an upgraded version proposed by Chang *et al*. (2003) find optimal routes on raster maps. The shortest path they find, however, is not identical to the optimal one. In the presence of many obstacles, the algorithms determine a route containing so many turning points and course alterations that no navigator would follow it.

In relation to the application of evolutionary computation of route planning, as proposed by many researchers, such as Ito *et al*. (1999), Smierzchalski and Michalewicz (2000) and Cheng *et al*. (2006), these research works evolve optimum routes for specially designed gene coding techniques and genetic operation. They mainly consider collision avoidance of static and dynamic objects but avoid multiple-goal situations in route planning during mid-distance coastal navigation. Apart from this, in the generation of the initial population of candidate routes, a random approach is taken and hence, the quality/efficiency is comparatively worse, making it hard to converge to the required high-quality solution. In this research, through examination of contributions in previous work, the benefits of GIS's spatial analysis and geometric computational capability are leveraged to improve the quality of candidate initial route population and improve the fitness function, gene coding and genetic operation in the genetic algorithm's route optimization.

3. PLANNING THE BEST ROUTE.

Route design is one of the skills that a navigator needs to grasp. The STCW 78/95 convention sets the appropriate standards that seafarers have to observe for route design. Route design is application-based, cross-domain and encompasses a large amount of complicated content. This is an integration and application of knowledge in subjects such as navigation, meteorology and geography. The best route, as described, would utilize known information (such as nautical charts, weather conditions, sailing directions, notices to mariners). The route can be pre-planned on a nautical chart (paper chart/electronic chart) and this route should be able to guarantee safety as well as the shortest sailing time, while being economical. The best route is not merely the route that is the shortest. The best route is constructed by choosing waypoints, in which different combinations of waypoints provide a large number of route lines that may be selected. In the planning of a route line, there are several main factors that need to be considered:

- Weather conditions: prevailing winds in the navigation area, monsoon, tropical cyclone, etc. In consideration, areas with non-favourable weather conditions should be avoided, while favourable conditions for the ship should be taken advantage of.
- Sea conditions: currents, wind, waves, etc. In consideration, areas with non-favourable sea conditions should be avoided.

- Obstacles: for example land, islands, shallow areas, reefs and water depth restricted areas. This refers to natural obstacles that endanger the safety of the ship's navigation, and the ship should be prohibited from entering these areas.
- Positioning conditions: routes should be planned to be near areas that would be beneficial to position fixing, to ensure the correct position of the ship and avoid the occurrence of deviations.
- Avoidance conditions: route planning should avoid entering narrow waters that restrict manoeuvrability, to avoid situations whereby restrictive water navigation makes collision avoidance difficult.
- Condition and status of the ship: includes stability, equipment, operational limitations, propeller and steering system reliability.
- Characteristics of goods (especially dangerous cargo), the distribution of goods onboard the ship, load and lashing.
- Crew's comfort level.
- Threatening areas of navigation: the area should not cause immediate danger and can be navigated into, but will increase the risk in the journey. Examples are pirate-prevalent areas, fishing areas, military exercise areas, fog-prone areas and floating ice areas. Ships should avoid entering such areas and any entry made should be as limited as possible.
- No-sail zones: man-made obstacle areas designated due to set national policy or political factors.
- Designated navigation zones: zones that need to be navigated through to satisfy navigational safety, for example, traffic separation schemes.
- Avoiding excessive turning points and overly large steering angles: the planned route line should be feasible and in line with sea navigation practice.
- The less the total mileage/sailing duration/fuel consumption, the better. This is to fulfil economy requirements.

We note that the factors that need to be accounted for in route planning are many and dynamic. It follows that planning the best route necessitates the goal that these conditions be fulfilled while, at the same time accounting for navigational safety and economy. To fulfil this goal, the route planning quality and efficiency would be enhanced if a multi-objective decision support system were available to assist in route planning to add to the knowledge and experience of the navigator. In the following sections, this research uses these factors as the basis of reference to design a model and system that supports automated obstacle avoidance route planning with multi-objective strategy support, and which satisfies sailing safety and economy requirements.

4. GIS IN INITIAL ROUTE POPULATION GENERATION.

Route planning is an important function of ECDIS and there are many ways to realize this. However, these are mostly related to GIS, where system implementation can be directly developed on the GIS platform or GIS's COM component can be embedded in a general information system to achieve the functions of ECDIS. This research uses embedded GIS as the development platform and Mercator projection as the map display and the basis of measurement. Through the strong capabilities of GIS's spatial information management, spatial analysis and geometric computation, the presence of obstacles is detected and candidate routes are automatically generated, providing the basis of initial population calculation for evolutionary computation. The role of GIS in this research and its execution procedures are detailed below.

4.1. *Spatial data processing*. Section 3 has mentioned many spatial objects that are related to route design. Some of these objects aid navigation, while others inhibit navigation. Route planning is faced with the management and analysis of these spatial objects, whereby the purpose is to avoid spatial obstacles, while making best use of beneficial spatial objects. GIS is used as the ECDIS development platform so that its spatial information storage, processing and analysis capabilities can be utilized, thereby reducing possible errors due to manual interpretation, especially when faced with near-coastal navigation environments. In particular, these environments contain multiple natural or artificial spatial objects that vary in their degree of impact. One-by-one analysis of these objects is time-consuming and hence, GIS's spatial information management and spatial analysis capabilities can be used.

4.1.1. *Data processing*. The route itself is a line spatial object, and spatial information related to navigation is constructed by points, lines or polygons. Therefore, the processing and analysis involved concerns computations between these three primary objects:

- A point object can be an isolated reef, a wreck or a navigation aid. It has a certain impact range and a navigation route would typically avoid them.
- A line object could be, for example, shallow isobaths or submarine cables. They also have a certain impact range and a navigation route could avoid or cross them.
- Polygon objects can be many different types of things, such as obstacle borders or other natural/man-made designated outlines, with a defined impact range. Normally, routes would avoid or cross them.

As described above, every spatial object in fact has a certain impact range that will affect route planning. Therefore, we use buffer functionality in GIS to set the impact range. This may be set in two ways: one is to use the route line as the centre line and the safe distance to be maintained as the range of the buffer zone, which constitutes a route strip polygon (as shown in Figure 1). When route planning, the rest of the spatial objects will be maintained outside this route strip. This is a simple method, but the impact range of each object is different. Hence, using a standard buffer zone to define the interactional scope with objects is non-uniform: that is, some scopes will be too big and others too small. As a result, this research uses another method, that is: setting an individual impact range buffer zone for each object, expanding each spatial object into a polygon spatial object (as in Figure 2). As for the size of the impact range, this can be set through an attribute table that assigns different size settings based on the type of the spatial object. For instance, clear border obstacle areas can be given a larger buffer zone size, whereas unclear areas such as borders of fishing zones or weather phenomenon areas can be given smaller buffer zone sizes. Similarly, dangerous areas are given larger buffer zone sizes, while areas that do not affect safety can be given smaller buffer zone sizes. If newly created buffer zones overlap, the overlapping buffer zones can be united to form a new polygon (as shown by the buffer zone union of points in Figure 2).

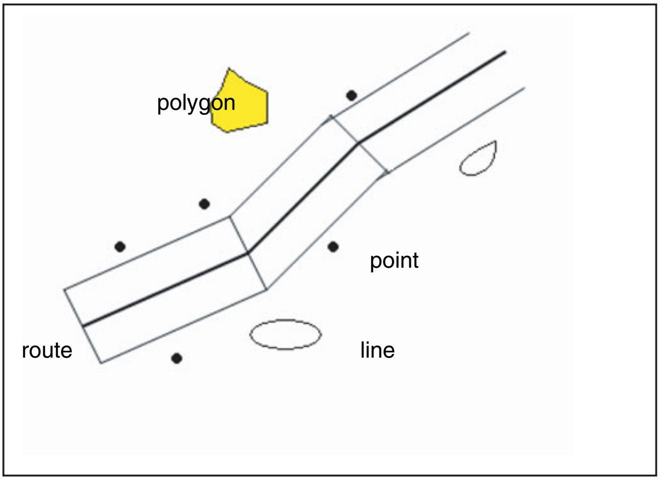

Figure 1. Route strip constructed using route line buffer zone.

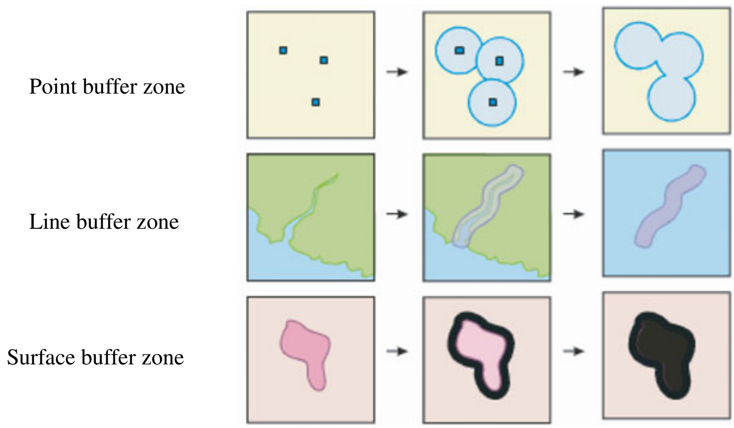

Figure 2. Generation of buffer zones for individual spatial objects.

4.1.2. *Spatial analysis*. After obtaining each spatial object's buffer zone, the next step is to use the overlap analysis functionality in GIS. Overlap analysis functionality includes intersection, containing, contained by, adjacent to and touched by. This research uses intersection to carry out testing. Through intersection testing of the route and obstacles, we can determine which spatial object intersects the route, the point of intersection and the length of intersection. This analysis determines whether the route planning will work and how to plan obstacle avoidance for the route.

4.2. *Route establishment*. Previously, we discussed how GIS is used to pre-process spatial data. With the processed data, the candidate routes are automatically generated using the spatial geometric computational functionality provided by GIS. The actual procedures are as follows:

4.2.1. *Minimum Bounding Rectangle* (*MBR*) *creation*. To find every buffer zone polygon's MBR, that is, the rectangle bounded by the polygon's maximum latitude

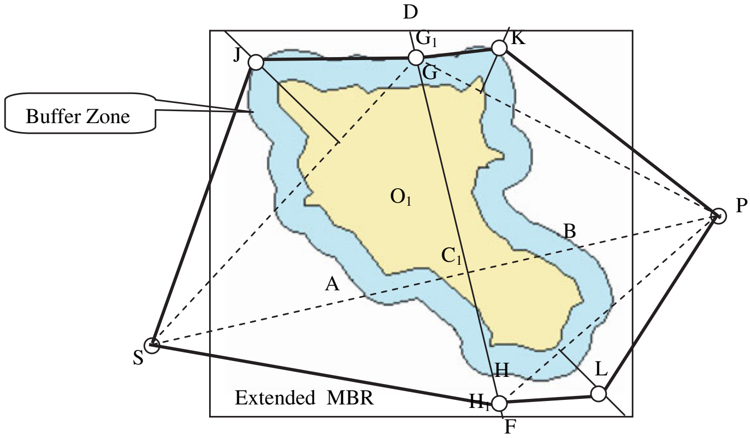

Figure 3. Obstacle avoidance route generation.

and left-most longitude as well as the polygon's minimum latitude and right-most longitude. In order to provide more feasible solutions, the rectangle can be expanded to a certain distance larger than the range of the MBR. This distance can be set according to the navigation plan and MBR size.

4.2.2. *Generating obstacle avoidance route*. As shown in Figure 3, when the ship navigates from starting point S to destination point D, it needs to be taken into consideration if the voyage will pass through the traffic separation scheme (a necessary part of the voyage that may not be changed). Thus, the start point referred to here can be the route's actual start point or the end point of the traffic separation scheme, regarded as the start point of a route segment. Similarly, the destination point can be the route's actual end point or the start point of some traffic separation scheme regarded as the route segment's end point. The route establishment steps are described below, and the Appendix shows the related C-language-like pseudo code:

- First, create a test line SP from S to P. If the test line does not pass through an obstacle area, then there is a straight line of navigation between the two points. If only considering navigation requirements, the straight line from S to P on the sea map is the shortest route.
- If passing through the obstacle area, then select polygon O_1 nearest to point S that intersects with the test line. Assume that at the intersection of SP and O_1, the nearest two points to the S point are Points A and B, where the mid-point of line AB is C_1. Next, create a straight line across C_1, perpendicular to line AB, intersection MBR at D and F, and intersecting O_1 at G and H.
- Randomly choose points G_1 and H_1 from line segment DG and line segment HF respectively, but ensuring that these two points cannot be within other obstacle areas. Then, separately create a line from point S to G_1 as SG_1 and S to H_1 as SH_1. These two line segments are two possible routes across two sides of the obstacle area but if part of the O_1 border is beyond the designated range, as indicated on the map, then only the route within the range area needs to be computed, or if the obstacle area has been divided into two route lines, only

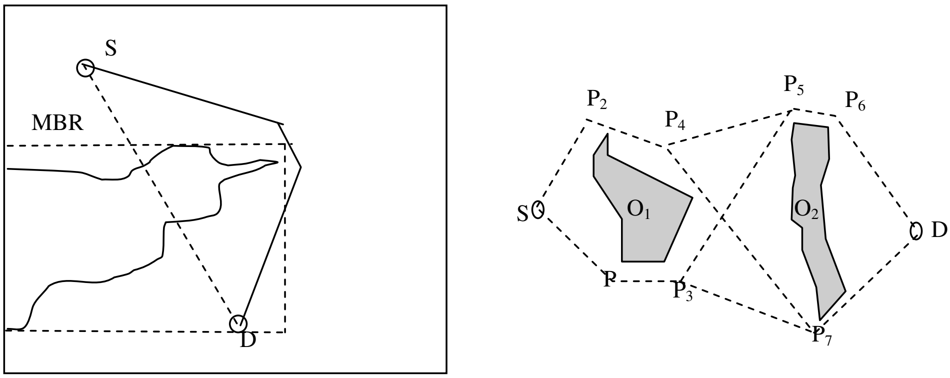

Figure 4. (Left) Unilateral navigation test. (Right) Bilateral multi-route line generation.

one side needs to be tested and the other can be ignored (as shown in Figure 4(Left)).

- Assume G_1 and H_1 as the end points. Then, recursively divide the line segment into two parts, repeatedly testing the second and third steps above, obtaining turning point J and line segments SJG_1 and SH_1 as the lines that pass through the first half segment of the obstacle area.
- Then, let G_1 and H_1 be the starting points and P the destination point. Similarly, recursively divide the line segment into two parts, repeatedly testing the second and third steps, obtaining turning points K, L, and line segments G_1KP and H_1LP as the lines that pass through the second half segment of the obstacle area.
- When the generated route line does not intersect with the current obstacle area, it means that it has completely bypassed the obstacle area. After connecting every waypoint, the two route lines SJG_1KP and SH_1LP can be obtained.
- In order to provide the initial population number of routes set by the genetic algorithm, we can set the initial population as n and the system will work using the settings in a repeat execution of the previous six steps.

The route constructed using the above steps is not the shortest route nor the best route but can provide the genetic algorithm for the initial population as a reference when it finds the best route, as the basis of evolution.

4.2.3. *Multiple route generation.* During route generation, when faced with new obstacle objects, separately record two possible route lines. As shown in Figure 4 (Right), when a new obstacle object O_1 is faced at point S, there is a need to simultaneously create two route lines SP_1 and SP_2, separately bypassing obstacle object O_1, and then separately continue testing with P_1 and P_2. If faced with obstacle object O_2, then two routes are needed to fork from the original two route lines. Specifically, P_3 divides to P_5 and P_7, P_4 divides to P_5 and P_7, forming four lines. Loop continuously until end point D can be visualized. Hence, when there are n navigation obstacle objects, 2^n routes need to be created and finally, the shortest one is selected from the created routes.

4.3. *Route Simplification.* Since many waypoints could be created during the route generation process and thus cause unnecessary turnings, it is necessary to simplify the route and remove redundant waypoints. In this research, we used the

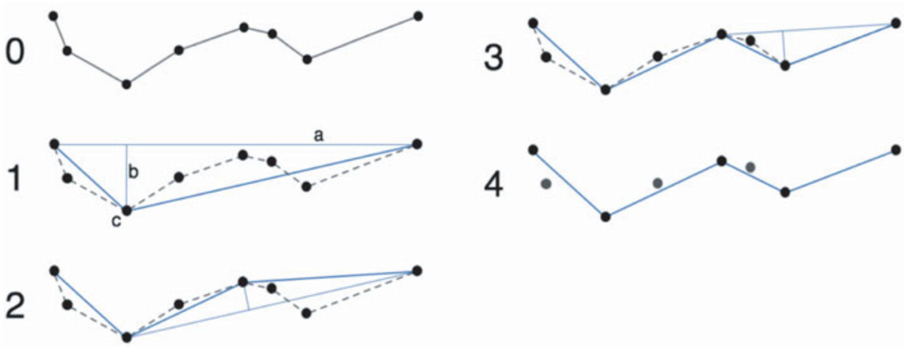

Figure 5. Diagram showing route simplification using the Douglas-Peucker algorithm (By Jitse Niesen).

Douglas-Peucker algorithm (Douglas and Peucker, 1973) to simplify routes (as shown in Figure 5). The threshold distance value for waypoint removal can be user defined but it needs to be noted that after simplification, the route must not cross over obstacle areas, otherwise the original route is to be maintained. After the simplification process, the original route SJG_1KP can be simplified to SJKP. Line segments belonging to the traffic separation scheme cannot be simplified.

5. GA ON AUTOMATIC ROUTE PLANNING MULTI-OBJECTIVE OPTIMIZATION.

This research targets a few important areas of GA including gene coding, fitness function design, selection operation, crossover operation, mutation operation and the control of parameter choice for revising route planning suitably. This is to enable the capability to optimize multi-objective route planning.

5.1. *Description of genetic algorithm.* A genetic algorithm (GA) is a stochastic optimization algorithm based on the principles of natural selection and natural heredity. With the aid of genetics and the evolutionary theory of Charles Darwin (Back, 1996), the whole problem space is searched randomly and in parallel to get the optimum solution. A genetic algorithm employs a population instead of a unit to search for the optimum solution and this entails the feature of parallel algorithms. In the evolutionary process of every generation, genetic algorithms carry out the same steps, such as reproduction, crossover, mutation and so on. Only the fitness function of the question itself is used. It does not need other preconditions and auxiliary information. Because there are fewer restrictions on fitness function and constraint conditions, and because the search range covers the whole independent variable space, optimum solutions can be obtained with a greater probability of iteration. Genetic algorithms use the random transfer rule, which can effectively solve complicated nonlinear problems for optimization in real-world applications.

5.2. *Route gene coding.* This research uses a series of waypoints to represent a line. Each waypoint can be represented as a gene and a combination of genes (waypoints) represents a chromosome (or a route). In order to represent this arrangement of the gene coding of the route, we use the Double Linked List data structure to represent a series of waypoints that form a route (as shown in Figure 6). The number of nodes in each route is different. Each waypoint node records the waypoint's

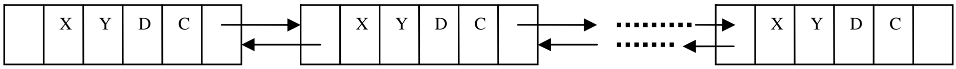

Figure 6. Representing a route's Double Linked List data structure. Number of nodes is not fixed.

longitude and latitude coordinates and the distance (D) and course (C) from the waypoint to the next waypoint. In relation to distance and course calculation, this research uses the rhumb line distance and course calculation method for Mercator Sailing in Hu *et al*. (2005) as the basis for the written program that pre-calculates the two variables and saves them in each node's field (except destination node). This enables rapid assessment of the fitness function. Compared to the traditionally used binary coding method of GA, this method has a clear significance for sea navigation as it facilitates calculation, operation and incorporates sea navigation domain knowledge. Furthermore, it reduces gene code and decoding complexity and improves calculation efficiency. Chung and Perez (1997) have performed a theoretical analysis of an approach similar to ours, which proposes the advantages of using a unified representation for the encoding method and the problem it is desired to solve.

5.3. *Initialization of route population*. The quality of the initial solution of the genetic algorithm is a key factor that determines whether the solution can subsequently converge to the optimal solution. As described in Section 2, evolutionary computation on obstacle avoidance route generation has mainly adopted random generation for the initial population of candidate routes. In other words, a series of waypoints are randomly generated within a navigational area to represent a route. The number of waypoints is not predefined and the waypoints may be located on land or obstacle areas that cannot be navigated on. As a result, the initial population created for the genetic algorithm would produce a worse performance with respect to quality or execution efficiency. Furthermore, it would be difficult to converge to the required optimal solution. As described in Section 4, after GIS generates higher quality initial candidate routes, the genetic algorithm can then flexibly process and perform optimization, effectively enhancing solution efficiency and quality.

5.4. *Elimination of obstacle avoidance routes through fitness function*. Fitness function is a function that represents an individual's survivability in a competition, generally used to perform optimization, while route planning needs to consider satisfying a number of bounding conditions and is a type of multi-objective optimization problem. In this research, through the use of a linear weighting technique, we converted the multi-objective problem to a single objective problem. The method is as follows: according to the priority level of f_j of each objective, extract a set of weighted coefficients $\lambda_j \geqslant 0$, $\Sigma\lambda_j = 1$, execute the evaluation function, $\mathrm{h(f)} = \Sigma\lambda_j f_j$, and then request a corresponding single objective plan: $F = \min h(f) = \min (\Sigma\lambda_j f_j)$ optimal solution set. Here, each objective f_j first needs to be normalized and, for items that do not satisfy the constraint conditions, the penalty function must be used to add the penalty value for violation of the constraint condition to the cost value. Similarly, for conditions that are beneficial toward navigation safety and economy, add a lower cost value. As there are a significant number of constraint conditions, subjective factors may be introduced when using the penalty function, such as function design and weight selection. Therefore, this research pre-filters routes that go across obstacle areas (man-made or natural). We assess three items only, the distance; navigable areas carrying a threat; and areas that benefit navigation. This changes the problem into a weak constraint multi-objective optimization problem. On the basis of the above considerations, the objective function is created as follows:

$$F=\min\left(\lambda_1 f'_1+\lambda_2 f'_2+\lambda_3 f'_3\right) \tag{1}$$

f'_1, f'_2, f'_3 are cost values after normalization. The method used in this research is: $f'_j=(f_j-\min f_j)/(\max f_j-\min f_j)$, where f'_1 is the value for the total voyage, which is a sum of the point distances (the lower, the better) and λ_1 is the weight value of the voyage cost value. f'_2 is the total voyage value for passing through threatening areas. Its value can be obtained using GIS's intersection functionality and then summed (the lower the sum, the better). Here, λ_2 is the penalty weight value for passing through threatening areas, where the threatening areas can be of several types, for example: narrow waters, foggy areas, pirate-prevalent areas, congested areas and fishing areas. Different weights can be given to the different areas or a single weight value may be set. f'_3 is the total voyage value for passing through areas beneficial to navigational safety or economy (for example, smooth water flows, good weather and passing through correct traffic separation schemes). The value can be obtained similarly to f'_2 but the higher its value, the better. Hence, f'_3's conversion method is: $(\max f_3-f_3)/(\max f_3-\min f_3)$, where λ_3 is the weight value for passing through areas beneficial to navigation safety or economy. Obtaining the weight involves subjective decision-making as it is reliant upon the synthesis of expert views or user-defined settings derived after adjustments in numerous experiments.

5.5. *Route selection.* During evaluation, in order to preserve good routes to speed up computation convergence, as well as preventing the loss of the best route in the group, an elite preservation strategy is used so that the best route in the current group does not need to participate in selection, mutation and crossover operations and is directly copied to the next generation. The remaining routes are then sorted according to their cost values in descending order, where each route in the group has a serial number. We directly use the serial number to perform selection. In the method we assume that some group has n chromosomes (routes) and then order the serial numbers according to their cost values in descending order so that the serial numbers are ordered as n, $n-1$, $n-2$, …, 2, 1. Then, we divide the serial numbers by n to obtain serial values: 1, $(n-1)/n$, …, $2/n$, $1/n$, using this to represent the probability that each chromosome (route) may be selected. Next, select using roulette selection. For the i^{th} chromosome (route), the selection probability Pi is $(n-i+1)/n$.

$$Si=\frac{\sum_{j=1}^{i} Pj}{\sum_{j=1}^{n} Pj} \tag{2}$$

When selecting a significant route, first produce a random number r between 0 and 1. The i^{th} chromosome's selection condition is:

$$\begin{cases} i>1, & if \;\; S_{i-1}<r<S_i \\ i=1, & if \;\; r<S_i \end{cases} \tag{3}$$

5.6. *Route's genetic operation.*

5.6.1. *Crossover operation.* Adopting a random single-point crossover method, the intersection location is chosen as a waypoint. Then, two chromosomes (routes) are randomly divided into two parts: the first part of the first chromosome is combined with the second part of the second chromosome and the remaining parts are combined. As a result, two new chromosomes (routes) are generated. It needs to be noted that the newly generated route should not pass through an obstacle area.

5.6.2. *Mutation operation.*

- *Create-disturbance operation.* After specifying the size of the disturbance's range, randomly change one of the node's coordinates in the route so that the changed coordinates are within the disturbance range specified.
- *Insertion operation.* Between two adjacent nodes, using the centre of the segment as midpoint and half the segment length as the radius, add a new node within the radius range of the midpoint.
- *Deletion operation.* Randomly delete a node on the route. The start point, destination point and points in designated areas cannot be deleted.

Each operation above should ensure that the route will not pass through non-navigable areas after mutation and that points belonging to the traffic separation scheme line segments may not run the mutation operation. Through a suitable mutation operation, we can ensure that the new route's quality will not deteriorate and fall into a local optimum.

6. SIMULATION RESULTS AND VERIFICATION.

6.1. *System setup.* In order to verify the results of our research with respect to automatic routing, we used Visual Basic.Net as the program development tool and ESRI's MapObject COM component as the platform for providing GIS functions. Spatial analysis and the display functions provided by the GIS component are seamlessly embedded into a general information system. This is integrated with the GA module in this research, customized to comply with the spatial decision support system required for automatic route planning in sea navigation. In this way, execution efficiency is further enhanced and the program I/O interface is more intuitive, making it easier and more immediate to conduct information exchange/integration with other navigation equipment or navigation information systems in the bridge (Beaubouef and Breckenridge, 2000). (The system operation interface can be seen in Figures 9 and 10.)

6.2. *Control parameter selection.* In the process of generating an obstacle avoidance route, parameters such as the size of the buffer zone required by GIS, the size of the MBR and the distance setting in the Douglas-Peucker algorithm are related to the navigation area size and the user's recognition of safety aspects. Therefore, the setting of these parameters can only be based on subjective decisions by users. Similarly, the setting of the weight parameters λ_1, λ_2, λ_3 of the fitness function is also subjective to user preference. Therefore, from user-defined settings, this research provisionally adopted journey distance weight (λ_1) as 0·6, threatening areas weight (λ_2) as 0·3 and beneficial areas weight (λ_3) as 0·1 for the simulations. We note that, with respect to GA's genetic operation parameter settings, as there was a large number of unfeasible solutions in the group, when the crossover rate and mutation

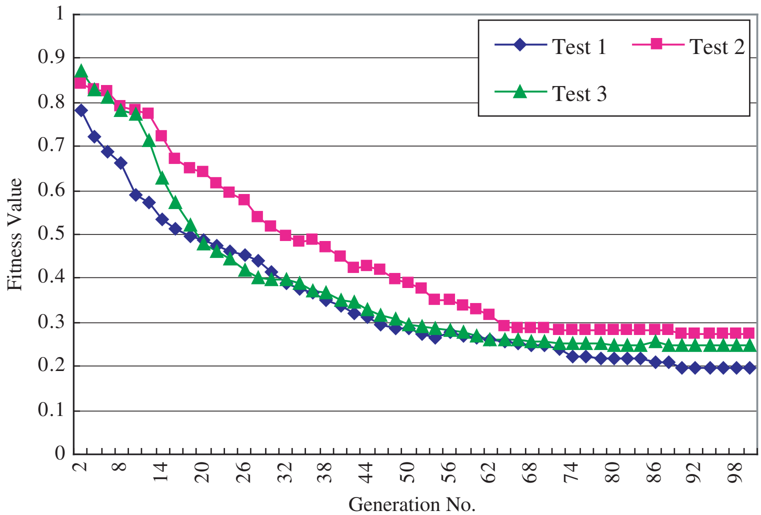

Figure 7. Evolution trends in best routes' fitness values.

rate are lower, the algorithm search space is smaller, making it harder to find a feasible solution. Conversely, if the crossover rate and mutation rate settings are too high, the computational time is increased. Therefore, this research used the Uniform Design Experimentation (Fang *et al.*, 2000) trial method to test the impact on the algorithm result of the four different factors of crossover rate, disturbance rate, deletion rate and insertion rate; each factor is assigned five levels. In the Uniform Design Experimentation method, one selects 20 sets of parameter combinations, sets the population as 100 and the evolution generation number as 100, for the experimentation. When the fitness function value of the best individual remains unchanged after eight generations of evolution cycles, the calculation is stopped. We display the changes in trend for three sets of experimental data (as shown in Figures 7 and 8). The parameter settings for each test are shown in Table 1. After a comprehensive consideration of the route's quality, ability to converge and computation time, Test 1 excels, regardless of whether individual or average group values are measured. Therefore, Test 1's settings were used as the GA's control parameters.

6.3. *Simulation results and analysis.* Figure 9 shows the single execution of the GIS processing technique described in this research on the planned candidate routes, each waypoint, and the patterns generated by the waypoints. Multi-objective planning has not been considered. From the figure, it can be observed that, through this method, two route lines Route N and Route S would be generated. From the perspective of choosing diversity or GA group origin diversity, this type of method is reasonable. This is because even if Route S is shorter, in consideration of multi-objective planning, if Route S passing by the sea will incur additional costs, then Route N can be considered as a candidate route. We also note that, in the figure, waypoint Q, waypoint R, waypoint T and waypoint P are evidently redundant waypoints but, through using the Douglas-Peucker Algorithm, only waypoint Q, waypoint R and waypoint T will be eliminated, while waypoint P is not eliminated. This is due to the random generation of each waypoint (random point selection along

Table 1. Test parameter settings.

Test	Crossover rate	Disturbance Rate	Insertion Rate	Deletion Rate
1	0·5	0·3	0·015	0·025
2	0·2	0·1	0·01	0·01
3	0·7	0·6	0·03	0·03

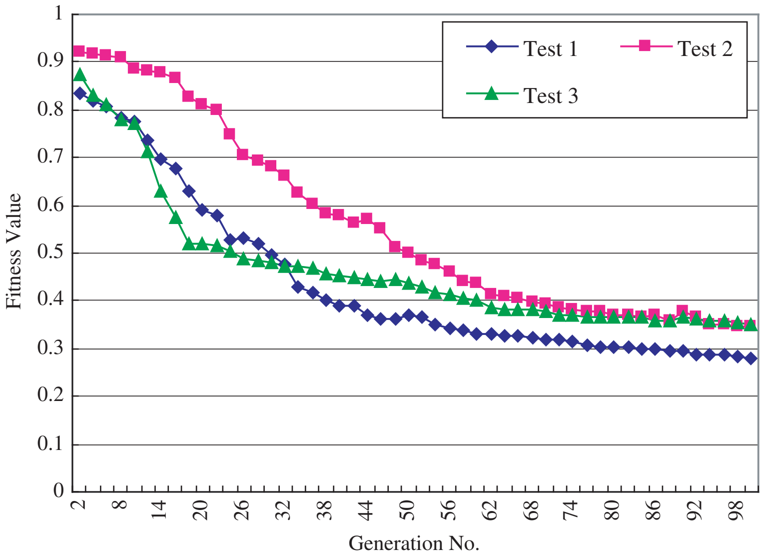

Figure 8. Mean population fitness value evolution trends.

the perpendicular bisector). As a result, it is possible to obtain a waypoint with undesirable qualities at random. Nevertheless, the randomness uncertainty range has been significantly reduced. Therefore, although the route generated through using the GIS geometric processing method is not necessarily the best route, the quality of the route generated will nevertheless be higher than that from a random search.

With respect to handling waypoint P, the near optimal route can be generated through the genetic operation of the second phase GA, for instance performing amendments using the deletion operation. Figure 10 shows the generation of the planned route from Honshu west bank to east bank by using a combination of GIS processing and GA optimization and the route generation process. Apart from considering route distance in this process, multi-objective factors such as threatening and beneficial navigation areas are also considered, to generate a recommended route. For the same experiment of generating the route from Honshu west bank to east bank, Figure 11 shows the fitness value evolution trends for route generation when using GIS processing and using random search (non GIS) processing. In the same scenario, we used an AMD Sempron2800 1.61 GHz processor, 1G memory and Windows XP Professional Edition environment to run simulations. After GIS pre-processing, every computation could be completed in between 10–15 seconds.

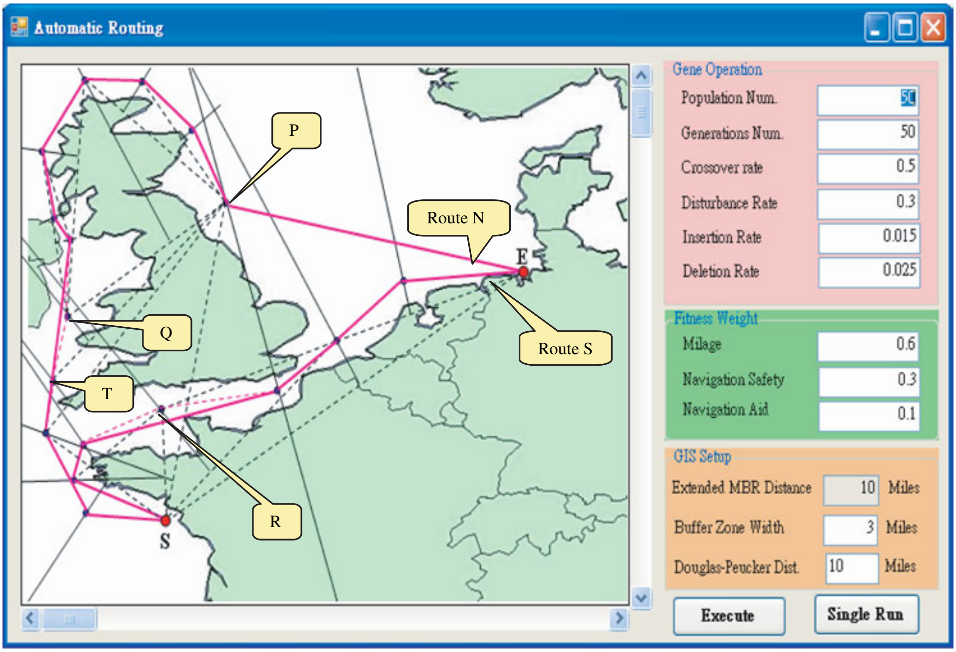

Figure 9. Using GIS single execution to plan alternative routes; multiple objectives not considered.

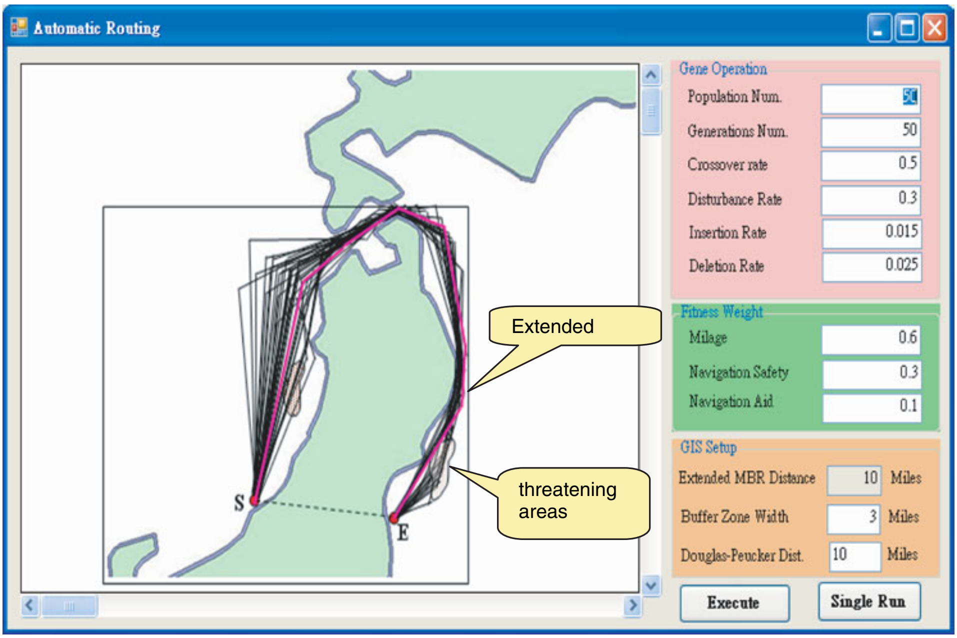

Figure 10. Automatic route generated from Honshu west bank to east bank, multi-objectives considered.

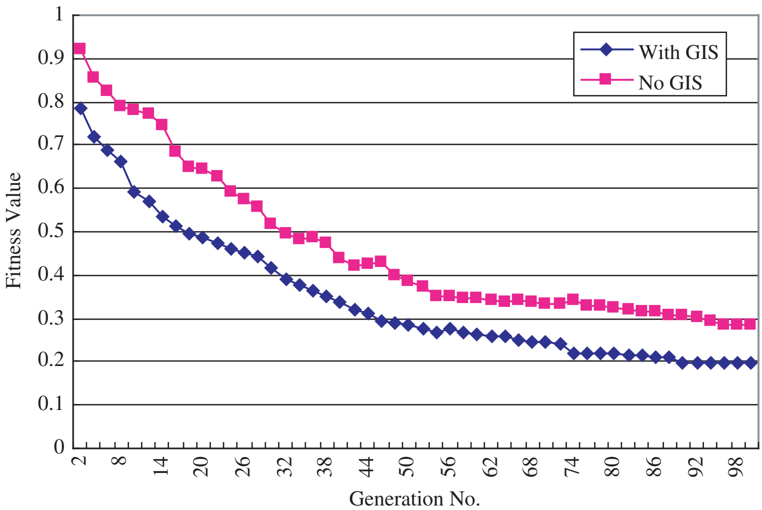

Figure 11. GA fitness value evolution comparison between GIS pre-processing and no GIS pre-processing.

In contrast, without GIS pre-processing and using random searches, the time taken is between 27–35 seconds. From the above experiments, we can observe that, after combining GIS pre-processing, both the quality and execution efficiency of the fitness value are improved.

7. CONCLUSION.

This research used a two-phase method. The first phase used GIS as the ECDIS development platform, using the GIS spatial information management, spatial analysis and geometric computation functionality to create automatic obstacle avoidance initial candidate routes and then to further simplify routes through the Douglas-Peucker Algorithm. After the GIS pre-processing, only candidate routes were generated, the optimal route was not obtained and the single objective considered was the shortest distance to avoid an obstacle. However, in order to better fulfil real navigation application requirements and achieve multi-objective route planning, the research then factored essential realistic considerations in route planning such as threatening areas, beneficial navigation areas and traffic separation schemes. Thus, the following phase employed a genetic algorithm (used to model biological evolutions), with quality candidate routes after GIS pre-processing as the basis, combined with specially designed functions for route planning including fitness function, gene coding and genetic operation to perform calculations. This method not only resolved the single objective limitation of geometric processing, but also avoided random searches and improved genetic algorithm efficiency. At the same time, safety and economy were considered, providing a reference to develop multi-objective optimized route planning strategies.

ACKNOWLEDGMENTS

It is appreciated that this research is subsidized by funding from the National Taiwan Ocean University (NTOU-RD972-01-03-34-01).

REFERENCES

Back, T. (1996) Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms, Oxford University Press.

Beaubouef, T. and Breckenridge, J. (2000) Real-world issues and applications for real-time geographic information systems (RT-GIS). *The Journal of Navigation*, **53**, 124–131.

Bijlsma, S. J. (2001) A Computational Method for the Solution of Optimal Control Problems in Ship Routing. *NAVIGATION, Journal of the Institute of Navigation*, 48, 145–154.

Bijlsma, S. J. (2002) On the Application of Optimal Control Theory and Dynamic Programming in Ship Routing. *NAVIGATION, Journal of the Institute of Navigation*, 49, 71–80.

Bijlsma, S. J. (2004) A Computational Method in Ship Routing Using the Concept of Limited Maneuverability. *The Journal of Navigation*, **57**, 357–369.

Bijlsma, S. J. (2008) Minimal Time Route Computation for Ships with Pre-Specified Voyage Fuel Consumption. *The Journal of Navigation*, **61**, 723–733.

Chang, K. Y., Jan, G. E. & Parberry, I. (2003) A Method for Searching Optimal Routes with Collision Avoidance on Raster Charts. *The Journal of Navigation*, **56**, 371–384.

Cheng, X. D., Liu, Z. Y. & Zhang, X. T. (2006) Trajectory Optimization for Ship Collision Avoidance System Using Genetic Algorithm. *IEEE OCEANS 2006 ASIA PACIFIC*, 1–6.

Chung, S. & Perez, R. (1997) Why is problem-dependent and high-level representation scheme better in a genetic algorithm? *Proc. of the 1997 ACM symposium on Applied Computing*, 239–246.

Coenen, F. P. (1980) Knowledge-based Collision Avoidance. *The Journal of Navigation*, **42**, 107–116.

Douglas, D. & Peucker, T. (1973) Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *The Canadian Cartographer*, 10(2), 112–122.

Dove, M. J., Burns, C. T. & Stockel, C. T. (1986) An Automatic Collision Avoidance and Guidance System for Marine Vehicles in Confined Waters. *The Journal of Navigation*, **39**, 180–190.

Fang, K. T., Lin, D. K. J., Winker, P. & Zhang, Y. (2000) Uniform Design: Theory and Application, *Technometrics*, 42, 237–248.

Hanssen, G. L. & James, R. W. (1960) Optimum Ship Routing. *The Journal of Navigation*, **13**, 253–272.

Harris, C. J., Hong, X. & Wilson, P. A. (1999) An Intelligent Guidance and Control System for Ship Obstacle Avoidance. *Proc. of The Institute of Mechanical Engineers Part I – Journal of Systems and Control Engineering*, *213*, 311–320.

Hayashi, S. & Kuwajima, S. (1991) A Strnding Avoidance System Using Radar Image Matching – Development and Experiment. *The Journal of Navigation*, **44**, 205–212.

Hu, J.-Q., Yang, Y.-S. & Li, T.-S. (2005) Accurate Calculation of the Course and Distance in Rhumb-Line (In Chinese). *Journal of Dalian Maritime University*, 31(2), 11–14.

Ito, M., Zhang, F. & Yoshida, N. (1999) Collision Avoidance of Ship with Genetic Algorithm. *Proc. of the 1999 IEEE International Conference on Control Applications*, 1791–1796.

Jones, K. D. (1978) Decision Making When Using Collision Avoidance System. *The Journal of Navigation*, **31**, 173–180.

Kao, S.-L., Lee, K.-T., Chang, K.-Y. & Ko, M.-D. (2007) A Fuzzy Logic Method for Collision Avoidance in Vessel Traffic Service. *The Journal of Navigation*, **60**, 17–31.

Lee, C. Y. (1961) An Algorithm for Path Connection and Its Applications. *IEEE Trans. Electron. Comput.*, EC-10, 346–365.

Lee, S. M., Kwon, K. Y. & Joh, J. (2004) A Fuzzy Logic for Autonomous Navigation of Marine Vehicles Satisfying COLREG Guidelines. *International Journal of Control Automation and Systems*, 2, 171–181.

Park, G.-K., Benedictos R. M., Lee, C.-S. & Wan, M.-H. (2007) Ontology-based Fuzzy-CBR Support System for Ship's Collision Avoidance. *Proc. of the 6th International Conference on Machine Learning and Cybernetics, Hong Kong*

Szlapczynska, J. & Smierzchalski, R. (2007) Adopted Isochrone Method Improving Ship Safety in Weather Routing with Evolutionary Approach. *International Journal Of Reliability Quality and Safety Engineering*, 14(6), 635–646.

Smierzchalski, R. & Michalewicz, Z. (2000) Modeling of ship trajectory in collision situations by an evolutionary algorithm. *IEEE Transactions On Evolutionary Computation*, 4, 227–241.

Zang, L.-H., Zhu, Q., Liu, Y.-C. & Li, S.-J. (2007) A method for Automatic Routing Based on ECDIS (In Chinese). *Journal of Dalian Maritime University*, 33(3), 109–112.

Zhu, X., Xu, H. & Lin, J. (2001) Domain and its model based on neural networks. *The Journal of Navigation*, **54**, 97–103.

Yang, S., Li, L. & Shi, C. (2007) Decision-Making Support System for Automatic Vessel Anti-Grounding and Anti-Reef (In Chinese). *Journal of Shanghai Maritime University*, China, 28(2), 14–20

APPENDIX

C-language-like pseudo code for generating automatic obstacle avoidance route lines.

```
Var SegmentList;   /* segment list for constructed route */
Var RouteList;     /* all possible routes */
Procedure   AutomaticRouting( )
    Var Start_Point, EndPoint;   /* route segment start and end points */
    Var Departure, Destination;   /* route start and end points */
    Var TrafficSeparationSchemeList;   /* create traffic separation scheme List */
    Put Traffic Separation Scheme that needs to be passed through into TrafficSeparationSchemeList;
    For I=0 to Number of Traffic Separation Scheme
      If I=0 then
        Start_Point=Departure;
        End_Point =(Number of Traffic Separation Scheme=0) ?
                    Destination : TrafficSeparationSchemeList[1+1].FromNode;
      else
        Start_Point=TrafficSeparationSchemeList[I].ToNode;
        End_Point =(I <Number of Traffic Separation Scheme) ?
                    TrafficSeparationSchemeList[I+1].FromNode : Destination;
        Put TrafficSeparationSchemeList[I] into SegmentList;
      GetRouteSegment(Start_Point, End_Point);
      From SegmentList, connect each segment into one or more routes by coordinate
         strings, and put them into RouteList;
      Use the Douglas Peuker method to perform generalization on all routes in
         RouteList, eliminating redundant nodes;
End of procedure

Procedure GetRouteSegment(S, E)     /* recursive function call */
    If   line segment SE does not pass through any obstacle object then
         Put line segment SE into SegmentList;
         Return;
    Else
      poly=the first obstacle object passed through by line segment SE;
      Extract the perpendicular bisector created by the two intersection points nearest to S point
      and intersecting poly.
      If   poly has already undergone intersection test OR poly's boundaries have exceeded the
      designated range then
        Along the perpendicular bisector, between the polygon unilateral border and MBR border,
        extract random point A;
        GetRouteSegment(S, A);
        GetRouteSegment(A, E);
      Else
        Along the perpendicular bisector, between the polygon bilateral border and MBR border,
        extract random points A, B;
        /* First, perform obstacle avoidance route plan for polygon unilateral border */
        GetRouteSegment(S, A);
        GetRouteSegment(A, E);
        /* Then, perform obstacle avoidance route plan for polygon bilateral border */
        GetRouteSegment(S, B);
        GetRouteSegment(B, E);
End of procedure
```